# Supporting Materials

# Data leakage management enables reliable machine-learning-based prediction of organic pollutant adsorption onto microplastics

Huatai Song [1], Jiaping Xu (✉)[1], Xiao Liang [2], Siyuan Li [1], Hao Ding [1], Yu Wang [1], Yiming Yao [1], Hongwen Sun (✉)[1]

1 MOE Key Laboratory of Pollution Processes and Environmental Criteria, College of Environmental Science and Engineering, Nankai University, Tianjin 300350, China

2 College of Software, Nankai University, Tianjin 300350, China

The supporting information (a total of 24 pages including the cover sheet) contains:

Text S1-S8

Figure S1-S6

Table S1-S6

## Contents



✉ Corresponding authors

E-mail: xujiaping@nankai.edu.cn (J. Xu); sunhongwen@nankai.edu.cn (H. Sun)

**Text S1. Data collection protocol.**

A data collection protocol was established to ensure data validity, given the diverse types of adsorbates and variations in experimental conditions across different studies. This protocol aims to prevent excessive outliers and poorly fitting isotherms from causing the model to capture noise.

The protocol requires that:

1) the sources of adsorption isotherms are limited to single adsorption systems, excluding competitive adsorption involving microplastics (MPs) and two adsorbates;

2) the data sources include linear, Freundlich, and Langmuir adsorption isotherms, as these three types are widely used in adsorption studies and have demonstrated good fitting performance;

3) isotherms with a coefficient of Determination ($R^2$) of 0.90 or higher are retained, and each adsorption isotherm includes at least five data points to guarantee data quality.

The data points were extracted using the WebPlotDigitizer, and were re-fitted in Origin. The $R^2$ values for both the re-fitted and original isotherms exceed 0.90, confirming the reliability of the digitizing tool.

**Text S2. Dataset construction.**

When constructing the dataset, it is necessary to consider the imputation of missing values. If the original literature provided specific surface area, glass transition temperature ($T_g$), and properties such as hydrogen to carbon ratio (H/C), oxygen to carbon ratio (O/C), and carbon content (C%), the data were extracted directly. For missing values of $T_g$, H/C, O/C, and C%, the average values of the same type of MPs reported in the literature were used for imputation. The Abraham parameters for organic pollutants (OPs) were obtained from the UFZ-LSER database.

The data points obtained from the adsorption isotherms are typically defined by two variables: the x-coordinate $C_e$ and the y-coordinate $Q_e$, where $C_e$ and $Q_e$ represent the equilibrium

concentrations of the target substance in the solution and on the MPs, respectively. The adsorption distribution coefficient ($K_d$), as shown in Eq. S1, is commonly used to describe the adsorption affinity of MPs for the target pollutants.

$$K_d = \frac{Q_e}{C_e} \tag{S1}$$

**Text S3. Cosine similarity.**

Cosine similarity is a widely used method for measuring the angular similarity between two non-zero vectors in the feature space of high-dimensional data. Its core idea is based on the direction of the vectors: if two vectors have the same or very similar directions, the angle between them is small, and the cosine value approaches 1, indicating high similarity; if two vectors are completely opposite (180°), the cosine value approaches -1; if two vectors are orthogonal (90°), the cosine value is 0, indicating no directional correlation. Specifically, cosine similarity is calculated as the dot product of two vectors divided by the product of their magnitudes, as follows in Eq. S2:

$$\text{cosine similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\| \mathbf{A} \| \| \mathbf{B} \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}} \tag{S2}$$

Where $\mathbf{A}$ and $\mathbf{B}$ are the two vectors being compared, and $n$ is the dimensionality of the vectors.

**Text S4. Comparison of different machine learning algorithms.**

Machine learning (ML) models can offer advantages in various applications, and different categories of ML algorithms have their own strengths and limitations for specific tasks. Therefore, for predicting the adsorption processes of MPs, we utilized 13 algorithms for comparison: A detailed description of each algorithm is provided below:

(1) Multiple linear regression (MLR) is one of the simplest and most intuitive prediction methods. It analyzes the relationship between a dependent variable and multiple independent variables, as shown in Eq. S3. (Darlington and Hayes, 2016).

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_n x_{in} + e_i \quad \text{(S3)}$$

Where $y_i$ is the i-th independent variable (output); $x_{i1}$,$x_{i2}$, ……$x_{in}$ are the i-th set of explanatory variables (inputs); $b_o$ is the intercept; $b_1$,$b_2$, ……$b_n$ are the regression coefficients; and $e_i$ is the random error term.

(2) Polynomial regression, like MLR, is a relatively simple regression analysis algorithm. It allows for higher-order nonlinear relationships between the independent and dependent variables, thereby fitting more complex data patterns, as shown in Eq. S4 (Magee, 1998).

$$y_i = b_0 + b_1 x_i + b_2 x_i^2 + \cdots + b_n x_i^n + e_i \quad \text{(S4)}$$

Where $y_i$ represents the i-th dependent variable (output); $x_i$ represents the i-th independent variable (input); $b_o$ is the intercept; $b_1$, $b_2$ , …… $b_n$ are the regression coefficients; $e_i$ is the error term associated with the i-th observation; and $n$ represents the degree of the polynomial.

(3) (4) (5) Ridge, Lasso, and Elastic Net regression models are extensions of linear regression designed to address issues such as multicollinearity. Ridge regression adds an L2 regularization term that controls the variance of coefficients in the presence of multicollinearity, enhancing prediction stability. Lasso regression, by minimizing the residual sum of squares subject to the constraint of the sum of absolute values of coefficients, enables feature selection and generates a parsimonious and interpretable model. Elastic Net combines the strengths of both approaches by incorporating L1 regularization for feature selection and L2 regularization to mitigate multicollinearity, making it particularly suitable for high-dimensional data with complex relationships. These models offer diverse solutions for multiple regression through their respective regularization strategies (Gabauer et al., 2024; Khan et al., 2019).

(6) Gradient boosting regression trees (GBM) is a model based on boosting techniques that improves prediction accuracy by sequentially training multiple weak learners (usually decision trees) (Wei et al., 2023). At each step, the GBM model fits a new weak learner to the negative gradient or residuals of the loss function with respect to the previous prediction. This

enables the model to gradually reduce prediction error and primarily lower the model bias, while variance can be controlled through regularization. However, this sequential training method is prone to overfitting. Therefore, it is necessary to prevent this issue by optimizing the model's hyperparameters (Kim and Park, 2022; Wang et al., 2021; Wei et al., 2019).

(7) Random forest (RF) is also a powerful trees-based ensemble learning method. By constructing multiple decision trees and combining their results, it enhances prediction accuracy and stability (Belgiu and Drăguţ, 2016). RF employs two sources of randomness: bootstrap sampling and feature random selection. These mechanisms ensure that each tree is trained on a different subset of data and features, which reduces the risk of overfitting and enhances the model's generalization ability. For prediction, RF aggregates the outputs of all decision trees through voting (for classification) or averaging (for regression) to produce the final results (Chen and Ishwaran, 2012; Sarica et al., 2017).

(8) Extreme gradient boosting (XGB) is an optimized implementation of GBM that introduces several enhancements over traditional GBM. By incorporating L1 and L2 regularization into the loss function and supporting early stopping, XGB effectively mitigates overfitting. Moreover, it optimizes the loss function using both first- and second-order gradients (e.g., a Newton-Raphson approximation), which significantly improves training speed and model stability. XGB is particularly well-suited for handling large-scale structured data (Kim and Park, 2022; Pathy and Meher, 2020).

(9) Light gradient boosting machine (LGBM) was developed to address the longer learning time of XGB. By employing techniques such as histogram algorithms and Gradient-based One-Side Sampling, LGBM significantly reduces training time while maintaining high prediction accuracy. It is particularly effective in handling large datasets and high-dimensional features (Lv et al., 2024). LGBM optimizes the tree growth process by using a leaf-wise growth strategy instead of the traditional level-wise growth, allowing the model to handle large datasets more efficiently with lower memory usage. Although LGBM outperforms XGB in terms of speed and resource utilization, it may be prone to overfitting on small the dataset (Chen et al., 2023).

(10) Multilayer perceptron (MLP) is a type of neural network consisting of layers of artificial neurons, where each layer passes its outputs to the next. The output of each neuron is calculated by multiplying the inputs by learnable weights and applying a non-linear activation function (Miche et al., 2009; Vergara et al., 2015).

(11) Extreme learning machine (ELM) is an algorithm based on the MLP, with its input weights and hidden layer biases randomly initialized. After random initialization, the relationship between the hidden layer output and the output layer becomes linear. The output weights are then analytically obtained by computing the pseudoinverse of the hidden layer output matrix. Specifically, the output weights are derived by solving a least-squares problem using this pseudoinverse, enabling a fast and efficient training process (Miche et al., 2009; Vergara et al., 2015).

(12) Support vector regression (SVR) performs regression by mapping input data to a high-dimensional feature space, where an optimal hyperplane is constructed to represent the nonlinear relationship between inputs and outputs (Chen et al., 2017; Panahi et al., 2020).

(13) Gaussian process regression (GPR) is a nonlinear regression method based on Gaussian processes. It defines a prior distribution over functions using a kernel function that specifies the covariance structure, enabling predictions within a Bayesian framework. GPR naturally balances model fit and complexity through marginal likelihood optimization. However, it has high computational complexity and slow inference speed, which limits its applicability to large-scale or high-dimensional datasets (Han et al., 2020;Wang et al., 2015).

**Text S5. Hyperparameter optimization.**

Hyperparameters are preset parameters before model training. Optimizing hyperparameters can lead to better performance (Baratchi et al., 2024). Two hyperparameter optimization methods are used in this study. The first method is grid search. Grid search finds the optimal hyperparameters by setting an initial grid of hyperparameters and traversing all possible hyperparameter combinations within the grid range at a fixed step size. Although grid search

typically requires extensive computation, it is straightforward and intuitive, making it easy to implement (Bischl et al., 2023). The second method is Bayesian optimization. Bayesian optimization hypothesizes the existence of a noisy black-box function that maps hyperparameters to model performance, constructing a surrogate model to approximate the objective function and iteratively optimizing the hyperparameters. This method excels in high-dimensional and complex objective function optimizations, effectively leveraging existing information to improve search efficiency. Although the computational process is relatively complex and may be influenced by the initial model and prior distribution, it can find good hyperparameter settings with fewer attempts, making it suitable for more complex model optimizations (Bischl et al., 2023; Xia et al., 2017). The predictive performance of the optimal models obtained via the two methods is compared, and the hyperparameter combination with better performance is selected as the representative. The specific hyperparameter search ranges and iteration details were shown in Table S1 and S2.

**Text S6. Model performance evaluation metrics.**

This study comprehensively utilizes three commonly used model performance evaluation metrics—$R^2$, root mean squared error (RMSE), and mean absolute error (MAE)—to describe model performance. $R^2$ primarily focuses on the overall fit of the model. RMSE is more sensitive to larger errors in predictions, amplifying the impact of errors, and emphasizes evaluating the stability of the model. In contrast, MAE gives equal weight to errors of any magnitude, making it more suitable for assessing the overall level of model error (Chicco et al., 2021; Safaei-Farouji and Kadkhodaie, 2022). The calculation methods for the three evaluation metrics are as shown in Eqs. S5–S7:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{S5}$$

Where $SS_{res}$ is the sum of squared residuals, and $SS_{tot}$ is the total sum of squares. The value of $R^2$ ranges from 0 to 1, with values closer to 1 indicating stronger explanatory power of the model over the data.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad \text{(S6)}$$

↔

$$MAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{n} \quad \text{(S7)}$$

Where $n$ is the number of data points, $\hat{y}_i$ is the predicted value, and $y_i$ is the observed value. Lower values of RMSE and MAE indicate higher prediction accuracy and better model performance.

**Text S7. Experimental validation.**

In the experimental validation process, polyethylene (PE), which had the largest proportion in the dataset (26.0%) from our previous study (Zhao et al., 2020), polyamide (PA), which also had a large proportion (14.5%), and thermoplastic polyurethane (TPU), which had the smallest proportion (0.7%) were used as sorbents. These selections were made to assess the model's ability to extrapolate to data-rich and data-poor MPs, respectively. Three contaminants—diethyl phthalate, naphthalene, and phenanthrene—were chosen as adsorbates.

Batch equilibrium experiments were performed with these substances by setting a varying log$C_e$ range and an experimental temperature of 25 ℃. Adsorption isotherms were fitted using Freundlich models, resulting in eight isotherms, each covering seven or more data points, with $R^2$ above 0.90 (Table S3).

**Text S8. Shapley additive explanations.**

Due to the black-box nature of ML models, their internal mechanisms and interpretability are commonly limited (Hong et al., 2016). In environmental research, high prediction accuracy alone is insufficient, methods that can reasonably explain the results are also necessary. Shapley additive explanation (SHAP) is one of the most widely used methods for model interpretation (Zhang et al., 2023). It is based on the Shapley value concept from game theory and provides a framework for fairly attributing the contribution of each feature to the prediction results (Lundberg and Lee, 2017). The formula for calculating Shapley values is as follows in Eq. S8:

$$g(x') = \varphi_0 + \sum_{M}^{j=1} \varphi_j \quad \text{(S8)}$$

where $g(x')$ is the model output, $\varphi_0$ is a constant term that explains the model (i.e., the predicted mean of all training samples), and $\varphi_j$ is the Shapley value representing the contribution of the j-th feature.

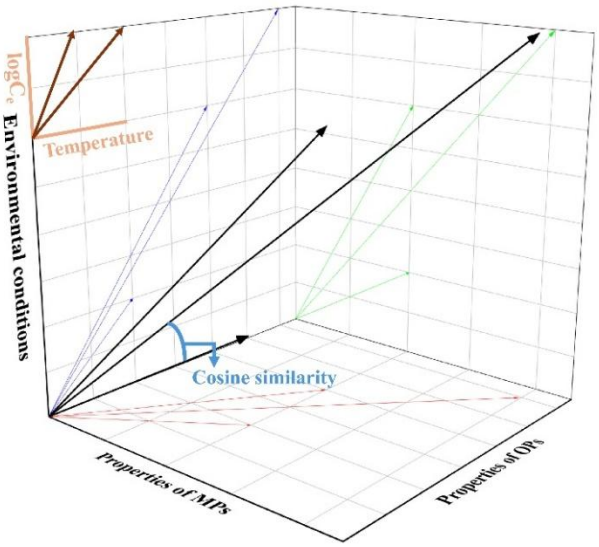


**Figure S1.** Illustration of feature vectors for adsorption data in a high-dimensional space.

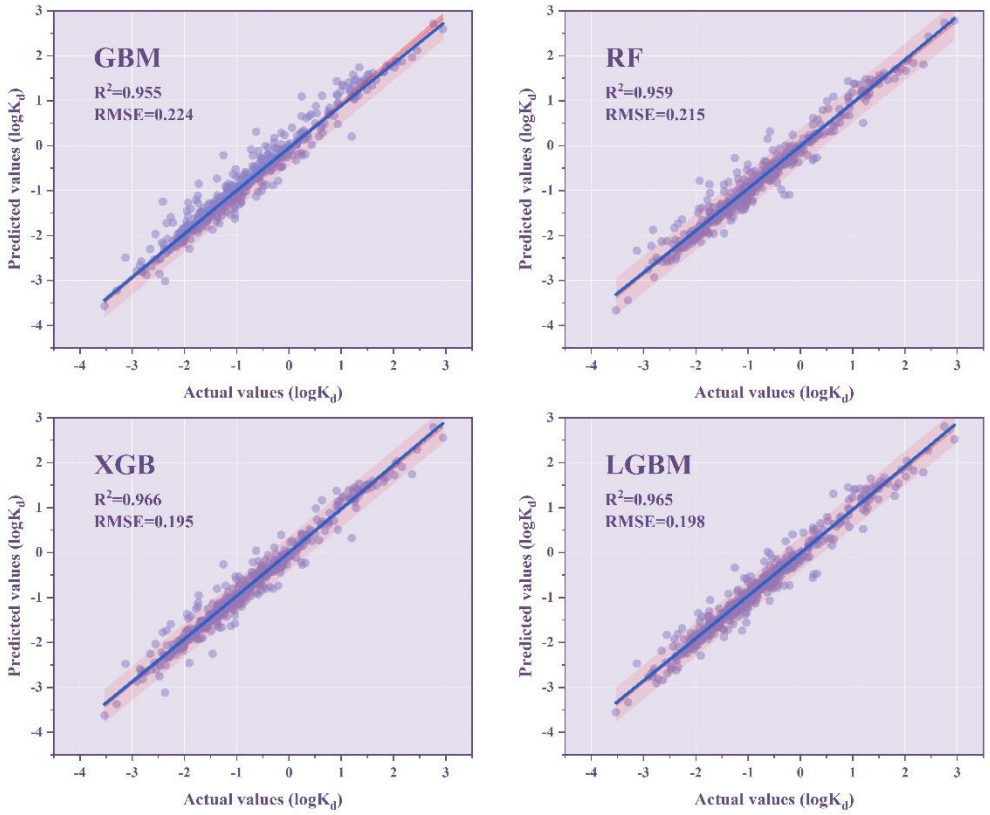


**Figure S2.** Testing set prediction results of four tree-based models without data leakage management (DLM).

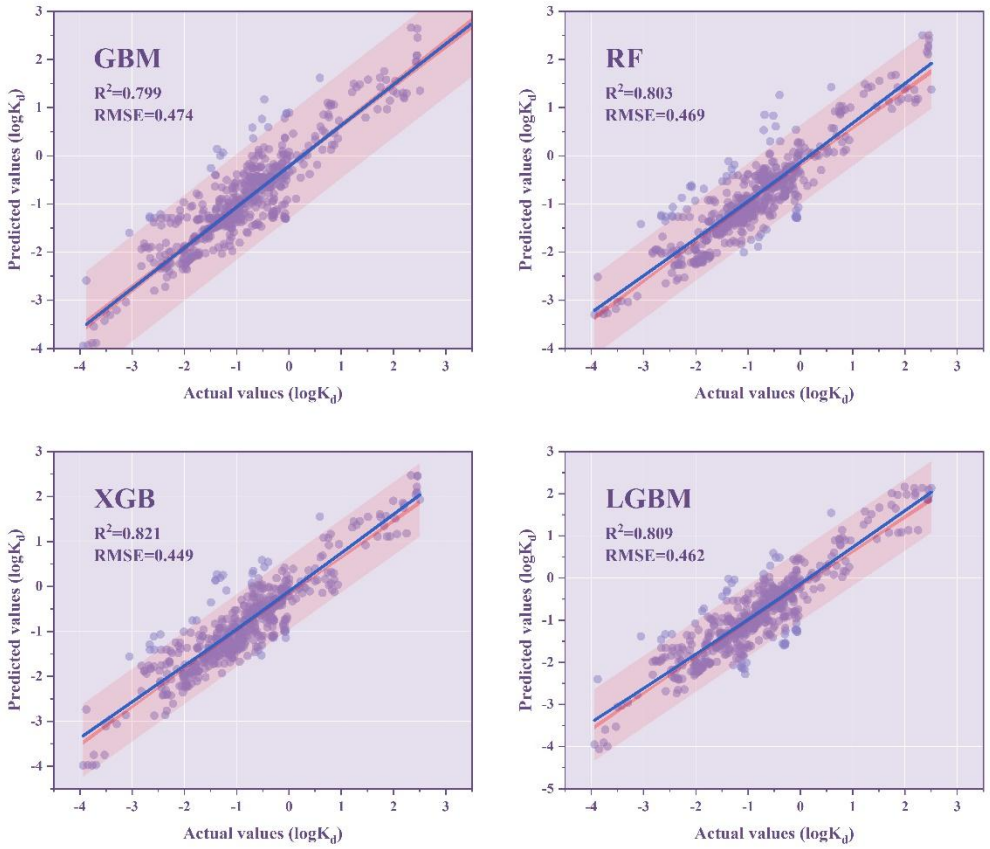


**Figure S3.** Testing set prediction results of four tree-based models with DLM based on isotherms.

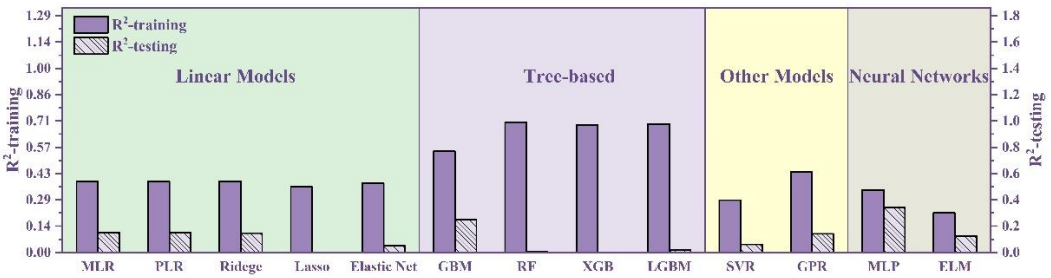


**Figure S4.** Prediction performance of different models on training and testing set with DLM based on studies.

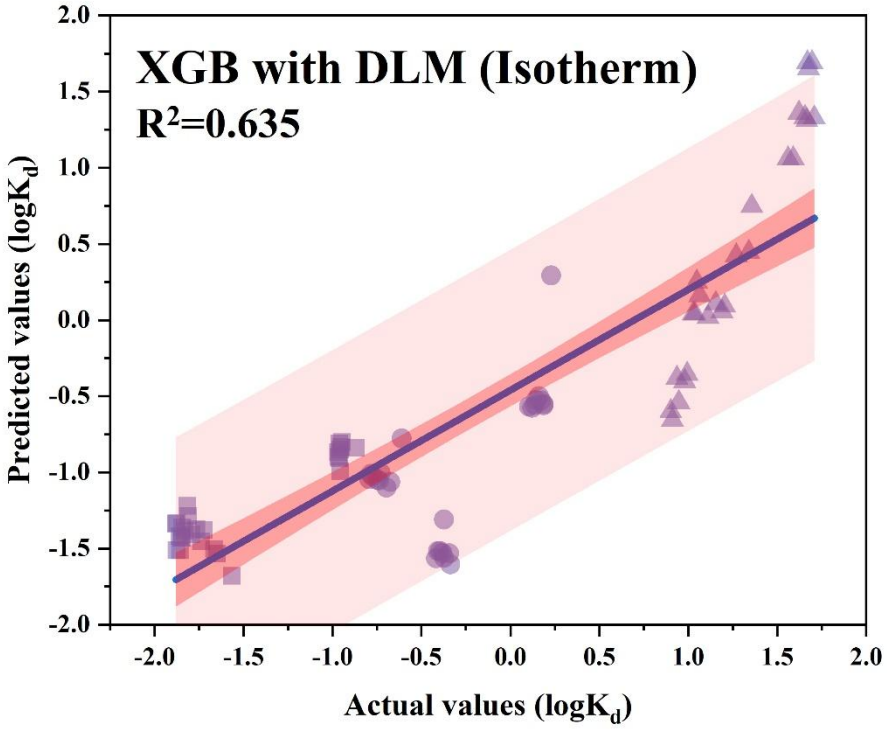


**Figure S5.** Experimental validation performance of the XGB model trained with isotherm-based DLM.

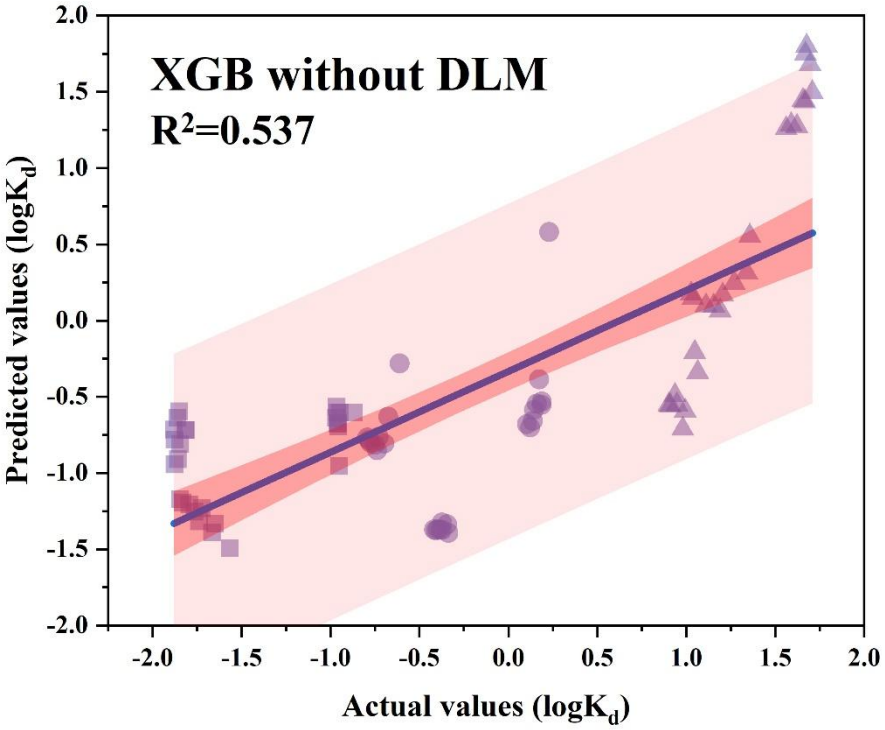


**Figure S6.** Experimental validation performance of the XGB model trained without DLM.

**Table S1**. Grid search hyperparameters, search ranges, and iteration settings.

| Model | Hyperparameters | Search Range | Candidates | Total Fits |
|---|---|---|---|---|
| MLR | None | - | 1 | 10 |
| PLR | degree | [1, 2, 3, 4, 5] | 5 | 50 |
| Ridge | alpha | [0.1, 1, 10, 100] | 4 | 40 |
| Lasso | alpha | [0.1, 1, 10, 100] | 4 | 40 |
| Elastic Net | alpha,l1_ratio | alpha: [0.1, 1, 10, 100]; l1_ratio: [0.1, 0.5, 0.9] | 12 | 120 |
| GBM | n_estimators, learning_rate, max_depth, min_samples_split, min_samples_leaf | 3 values each (243 combos) | 243 | 2430 |
| RF | n_estimators, max_depth, min_samples_split, min_samples_leaf, bootstrap | 4 values each (288 combos) | 288 | 2880 |
| XGB | n_estimators, max_depth, learning_rate, subsample, colsample_bytree | 3 values each (324 combos) | 324 | 3240 |
| LGBM | n_estimators, max_depth, learning_rate, num_leaves, subsample, colsample_bytree | 3 values each (972 combos) | 972 | 9720 |
| SVR | kernel, C, epsilon | kernel: 4 options; C: 4 values; epsilon: 4 values | 64 | 640 |
| GPR | alpha, kernel | alpha: 3 values; kernel: 3 values | 9 | 90 |
| MLP | hidden_layer_sizes, activation, solver, alpha, learning_rate | 3–4 values each (144 combos) | 144 | 1440 |
| ELM | n_hidden, activation | n_hidden: [10, 20, 50, 100]; activation: [sigm, tanh, lin] | 12 | 120 |

All models using grid search were tuned with 10-fold GroupKFold cross-validation on the training set (2146 samples).

**Table S2.** Bayesian optimization hyperparameters, search ranges, and iteration settings.

| Model | Hyperparameters | Search Space |
|---|---|---|
| PLR | degree | Integer(1, 5) |
| Ridge | alpha | Real(0.01, 100, 'uniform') |
| Lasso | alpha | Real(0.01, 100, 'uniform') |
| Elastic Net | alpha, l1_ratio | alpha: Real(0.01, 100, 'uniform'); l1_ratio: Real(0.01, 1.0, 'uniform') |
| GBM | n_estimators, learning_rate, max_depth, min_samples_split, min_samples_leaf | Integer or uniform distributions |
| RF | n_estimators, max_depth, min_samples_split, min_samples_leaf, bootstrap | Integer or categorical distributions |
| XGB | n_estimators, max_depth, learning_rate, subsample, colsample_bytree | Integer or uniform distributions |
| LGBM | n_estimators, max_depth, learning_rate, num_leaves, subsample, colsample_bytree | Integer or uniform distributions |
| SVR | kernel, C, epsilon | kernel: Categorical; C: Real(0.1, 100, 'log-uniform'); epsilon: Real(0.01, 0.5, 'uniform') |
| GPR | alpha, kernel__k1__constant_value, kernel__k2__length_scale | Log-uniform distributions |
| MLP | hidden_layer_sizes, activation, solver, alpha, learning_rate | Categorical or log-uniform |

Bayesian optimization models were also tuned with 10-fold GroupKFold cross-validation. The number of iterations (n_iter) was set to 50 for all models.

**Table S3.** Freundlich model parameters fitted from different adsorption experiments.

| **MPs** | **OPs** | $K_f$ | **N** | $R^2$ |
|---|---|---|---|---|
| PA | DEP | 0.014 | 0.977 | 0.996 |
| TPU | DEP | 0.111 | 0.996 | 1.000 |
| PA | NAP | 0.206 | 0.828 | 1.000 |
| TPU | NAP | 1.490 | 0.894 | 0.993 |
| PA | PHE | 8.872 | 0.781 | 0.991 |
| TPU | PHE | 33.207 | 0.884 | 0.980 |
| PA | CLO | 0.011 | 0.895 | 0.953 |
| TPU | CLO | 0.004 | 0.952 | 0.976 |

MPs: microplastics; PA: polyamide; TPU: thermoplastic polyurethane.

OPs: organic pollutants; DEP: diethyl phthalate (99%); PHE: phenanthrene (95%); NAP: naphthalene (98%); CLO: clothianidin (98%). All target chemicals were purchased from Shanghai Macklin Biochemical Co., Ltd. (China).

**Table S4.** Model performance of different models without DLM and the hyperparameter optimization method used.

| Model | Performance | | | | | | HPO method |
|---|---|---|---|---|---|---|---|
| | Training | | | Testing | | | |
| | $R^2$ | RMSE | MAE | $R^2$ | RMSE | MAE | |
| MLR | 0.527 | 0.792 | 0.614 | 0.494 | 0.751 | 0.584 | Both |
| PLR | 0.780 | 0.540 | 0.414 | 0.732 | 0.546 | 0.414 | Both |
| Ridge | 0.527 | 0.792 | 0.614 | 0.496 | 0.749 | 0.583 | BYS |
| Lasso | 0.464 | 0.844 | 0.659 | 0.448 | 0.784 | 0.619 | BYS |
| Elastic Net | 0.465 | 0.843 | 0.659 | 0.448 | 0.784 | 0.619 | BYS |
| GBM | 0.995 | 0.081 | 0.057 | 0.955 | 0.224 | 0.150 | GS |
| RF | 0.994 | 0.088 | 0.053 | 0.959 | 0.215 | 0.136 | GS |
| XGB | 0.999 | 0.045 | 0.031 | 0.966 | 0.195 | 0.124 | GS |
| LGBM | 0.996 | 0.077 | 0.051 | 0.965 | 0.198 | 0.132 | GS |
| SVR | 0.986 | 0.136 | 0.090 | 0.957 | 0.220 | 0.144 | GS |
| GPR | 0.873 | 0.411 | 0.297 | 0.844 | 0.417 | 0.305 | Both |
| MLP | 0.978 | 0.173 | 0.121 | 0.929 | 0.281 | 0.188 | GS |
| ELM | 0.704 | 0.627 | 0.494 | 0.659 | 0.616 | 0.476 | GS |

Note: BYS, Bayesian optimization; GS, grid search; HPO: hyperparameter optimization

**Table S5.** Model performance of different models with isotherm-based DLM and the hyperparameter optimization method used.

| Model | Performance | | | | | | HPO method |
|---|---|---|---|---|---|---|---|
| | Training | | | Testing | | | |
| | $R^2$ | RMSE | MAE | $R^2$ | RMSE | MAE | |
| MLR | 0.556 | 0.768 | 0.600 | 0.339 | 0.858 | 0.642 | Both |
| PLR | 0.556 | 0.768 | 0.600 | 0.339 | 0.858 | 0.642 | Both |
| Ridge | 0.554 | 0.770 | 0.603 | 0.349 | 0.852 | 0.640 | GS |
| Lasso | 0.493 | 0.821 | 0.644 | 0.282 | 0.895 | 0.685 | BYS |
| Elastic Net | 0.555 | 0.769 | 0.601 | 0.342 | 0.857 | 0.643 | BYS |
| GBM | 0.992 | 0.102 | 0.070 | 0.799 | 0.474 | 0.341 | BYS |
| RF | 0.979 | 0.167 | 0.114 | 0.803 | 0.469 | 0.332 | BYS |
| XGB | 0.985 | 0.141 | 0.102 | 0.821 | 0.447 | 0.324 | GS |
| LGBM | 0.983 | 0.152 | 0.107 | 0.809 | 0.462 | 0.335 | BYS |
| SVR | 0.854 | 0.440 | 0.286 | 0.703 | 0.575 | 0.431 | GS |
| GPR | 0.863 | 0.426 | 0.309 | 0.694 | 0.584 | 0.411 | BYS |
| MLP | 0.874 | 0.408 | 0.304 | 0.625 | 0.647 | 0.482 | GS |
| ELM | 0.633 | 0.698 | 0.542 | 0.391 | 0.824 | 0.643 | BYS |

Note: BYS, Bayesian optimization; GS, grid search.

**Table S6.** Number of studies included in the training set using different DLM methods.

| **DLM methods** | **Studies included in training subset** |
|---|---|
| Without DLM | 82 |
| Study | 65 |
| Isotherm | 82 |

## References


Baratchi, M., et al., 2024. Automated machine learning: past, present and future. Artif. Intell. Rev. 57**,** 122. https://doi.org/10.1007/s10462-024-10726-1.

Belgiu, M., Drăguţ, L., 2016. Random forest in remote sensing: A review of applications and future directions. ISPRS J. Photogramm. Remote Sens. 114**,** 24-31. https://doi.org/10.1016/j.isprsjprs.2016.01.011.

Bischl, B., et al., 2023. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 13**,** e1484. https://doi.org/10.1002/widm.1484.

Chen, H., et al., 2023. Shield attitude prediction based on Bayesian-LGBM machine learning. Inf. Sci. 632**,** 105-129. https://doi.org/10.1016/j.ins.2023.03.004.

Chen, X., Ishwaran, H., 2012. Random forests for genomic data analysis. Genomics. 99**,** 323-329. https://doi.org/10.1016/j.ygeno.2012.04.003.

Chen, Y., et al., 2017. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. Appl. Energy. 195**,** 659-670. https://doi.org/10.1016/j.apenergy.2017.03.034.

Chicco, D., et al., 2021. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput. Sci. 7**,** e623. https://doi.org/10.7717/peerj-cs.623.

Darlington, R. B., Hayes, A. F., 2016. Regression analysis and linear models: Concepts, applications, and implementation. Guilford Publications.

Gabauer, D., et al., 2024. Estimating US housing price network connectedness: Evidence from dynamic Elastic Net, Lasso, and ridge vector autoregressive models. Int. Rev. Econ. Finance. 89**,** 349-362. https://doi.org/10.1016/j.iref.2023.10.013.

Han, J., et al., 2020. Prediction of winter wheat yield based on multi-source data and machine learning in China. Remote Sens. 12**,** 236. https://doi.org/10.3390/rs12020236.

Hong, H., et al., 2016. GIS-based landslide spatial modeling in Ganzhou City, China. Arab. J. Geosci. 9**,** 112. https://doi.org/10.1007/s12517-015-2094-y.

Khan, M. H. R., et al., 2019. Stability selection for lasso, ridge and elastic net implemented with AFT models. Stat. Appl. Genet. Mol. Biol. 18. https://doi.org/10.1515/sagmb-2017-0001.

Kim, C., Park, T., 2022. Predicting determinants of lifelong learning intention using gradient boosting machine (GBM) with grid search. Sustainability. 14**,** 5256. https://doi.org/10.3390/su14095256.

Lundberg, S. M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. Adv. Neural Inf. Process. Syst. 30. https://dl.acm.org/doi/10.5555/3295222.3295230.

Lv, Y., et al., 2024. Precipitation Retrieval from FY-3G/MWRI-RM Based on SMOTE-LGBM. Atmosphere. 15**,** 1268. https://doi.org/10.3390/atmos15111268.

Magee, L., 1998. Nonlocal behavior in polynomial regressions. Am. Stat. 52**,** 20-22. https://doi.org/10.2307/2685560.

Miche, Y., et al., 2009. OP-ELM: optimally pruned extreme learning machine. IEEE Trans. Neural Netw. 21**,** 158-162. https://doi.org/10.1109/TNN.2009.2036259.

Panahi, M., et al., 2020. Spatial prediction of groundwater potential mapping based on convolutional neural network (CNN) and support vector regression (SVR). J. Hydrol. 588**,** 125033. https://doi.org/10.1016/j.jhydrol.2020.125033.

Pathy, A., Meher, S., 2020. Predicting algal biochar yield using eXtreme Gradient Boosting (XGB) algorithm of machine learning methods. Algal Res. 50**,** 102006. https://doi.org/10.1016/j.algal.2020.102006.

Safaei-Farouji, M., Kadkhodaie, A., 2022. Application of ensemble machine learning methods for kerogen type estimation from petrophysical well logs. J. Pet. Sci. Eng. 208**,** 109455. https://doi.org/10.1016/j.petrol.2021.109455.

Sarica, A., et al., 2017. Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: a systematic review. Front. Aging Neurosci. 9**,** 329. https://doi.org/10.3389/fnagi.2017.00329.

Vergara, G., et al., Comparing elm against mlp for electrical power prediction in buildings. International Work-Conference on the Interplay Between Natural and Artificial Computation. Springer, 2015, pp. 409-418.

Wang, J., Hu, J., 2015. A robust combination approach for short-term wind speed forecasting and analysis–Combination of the ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts using a GPR (Gaussian Process Regression) model. Energy. 93**,** 41-56. https://doi.org/10.1016/j.energy.2015.08.045.

Wang, S., et al., 2021. Machine-learning micropattern manufacturing. Nano Today. 38**,** 101152. https://doi.org/10.1016/j.nantod.2021.101152.

Wei, X., et al., 2023. Analyzing of metal organic frameworks performance in CH4 adsorption using machine learning techniques: A GBRT model based on small training dataset. J. Environ. Chem. Eng. 11**,** 110086. https://doi.org/10.1016/j.jece.2023.110086.

Wei, Y., et al., 2019. Estimation of surface downward shortwave radiation over China from AVHRR data based on four machine learning methods. Solar Energy. 177**,** 32-46. https://doi.org/10.1016/j.solener.2018.11.008.

Xia, Y., et al., 2017. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. Expert Syst. Appl. 78**,** 225-241. https://doi.org/10.1016/j.eswa.2017.02.017.

Zhang, J., et al., 2023. Insights into geospatial heterogeneity of landslide susceptibility based on the SHAP-XGBoost model. J. Environ. Manage. 332**,** 117357. https://doi.org/10.1016/j.jenvman.2023.117357.

Zhao, L., et al., 2020. Sorption of five organic compounds by polar and nonpolar microplastics. Chemosphere. 257**,** 127206. https://doi.org/10.1016/j.chemosphere.2020.127206.